

REMARKS

This Response and Amendment is submitted in complete response to the Office Action mailed December 19, 2002 (hereinafter, the "Office Action"). The Specification has been amended to more clearly define the present invention; no new matter has been added. Claims 1-19 are currently pending in this application. Claims 1, 2, 6, 17 and 19 have been amended to better define the instant invention.

It is believed that no additional claims fees are due or owing in regard to the submission of this Response and Amendment and its attached and related papers. However, if such fees are deemed due, the Office is invited to contact the undersigned at the address and telephone number listed below. A Petition for a three (3) month extension of time together with appropriate fees \$930.00 has been being submitted contemporaneously herewith, a grant of the Petition is earnestly requested. Entry of this paper and all amendments and Remarks contained herein is earnestly solicited. Favorable action in response to this paper is earnestly requested, as in a grant of a U.S. Patent.

The paragraphs that follow in conjunction with the amendments to the application presented above are submitted in complete response to this points raised in the Office Action that require such action.

I. **Specification**

At paragraph 1 on page 2 of the Office Action, the Examiner indicates that certain language in Claim 17 lacks proper antecedent basis. In particular, the Examiner points to the language “the at least one second software package as having an antecedent basis.” In response to claim 17 has been amended to correct matters of form. Accordingly, it is earnestly requested that the Examiner reconsider claim 17 as amended.

At paragraph 2 on page 2 of the Office, the Examiner objects to the Abstract on the basis of word-count. A new, corrected abstract is being submitted herewith as a separate page.

Drawings

The drawings are objected to because of the following:

- a. Figure 1 has no label.
- b. In Figure 1, element text is illegible.

A proposed drawing correction has been submitted in reply to this Office Action.

Claim Rejections - 35 USC §112

At paragraph 5 on page 3 of the Office Action, the Examiner states that claim 17 recites the limitation “the at least one second software package” in lines 2-3, and that there is insufficient antecedent basis for this limitation in the claim. As noted above, claim 17 has been amended to depend from claim 15 to overcome this rejection.

At paragraph 6 on page 3 of the Office Action, the Examiner states claim 19 recites the limitation “at least one second software package” in lines 3-4, and that there is insufficient

antecedent basis for this limitation in the claim. Claim 19 has been amended to depend from claim 15 to overcome this rejection.

Claim Rejections - 35 USC §103

A. Claims 1-4, 8, 9, 11 13-15 and 18 - Gish and Cox:

Claims 1-4, 8, 9, 11, 13-15 and 18 are rejected under 35 U.S.C. 103(a) allegedly as being unpatentable over Gish in view of Cox et al. **This rejection is respectfully traversed.**

Regarding independent claims 1, 11 and 14, the Examiner states Gish teaches the claimed distributed server administration system except for disclosing specifically means wherein the client is an administrator and the execution of the at least one first software package (citing the abstract; col. 5, ls. 50-60; and col. 18, ls. 13-27). The Examiner contends the features of allowing the server to be administered by the client administrator via the electronic data network would have been an obvious modification of the system disclosed by Gish as evidenced by Cox et al.

Cox et al is cited as disclosing a system for distributed management of network elements including servers with means wherein the client is an administrator and the execution of the at least one first software package allow the server to be administered by the client administrator via the electronic data network (col. 3, ls. 54-55, 61-67; col. 8, ls. 49-55; col. 9, ls. 7-11) wherein programs are distributed to the client for the operation and management of a program that exists on a server.

Gish discloses an application composed of a client (front end) program which communicates utilizing a network with a server (back end) program. The client and server

programs are loosely coupled and exchange information using the network. The client program is composed of a User Interface (UI) and an object-oriented framework (Presentation Engine (PE) framework). The UI exchanges data messages with the framework. The framework is designed to handle two types of messages: (1) those from the UI, and (2) those from the server (back end) program via the network. The framework includes a component, the mediator, which manages messages coming into and going out of the framework (col. 5, ls. 31-44).

The Examiner relies in particular on the disclosure in Gish that a plurality of client computer code segments are stored on the server. Each client code segment is designed for transmission over the network to a client computer to initiate coupling. A plurality of server computer code segments are also stored on the server. A particular client code segment is selected, installed and executed in response to initiation of coupling via the network with a particular client utilizing the transmitted client computer code segment for communicating via a particular communication protocol to a corresponding server code segment on the server computer (col., 5, ls. 50-60).

The Examiner also relies on Fig. 5 in Gish. Fig. 5 illustrates how a preferred embodiment leverages Java to facilitate the establishment and implementation of server-centric policies. The client and server nodes communicate utilizing the web technologies in an Internet, Intranet or other network environment. The client node 500 contacts the server node 520 via HTTP with a request to execute an application. After authenticating the client 500, the server node 520 selects front 502 and back end 510 components based on the application definition list maintained at the server node 520. The server starts its back end process 510 and sends the front end program 502 to the client node via the Web technologies in an Internet, Intranet or other network environment. The client 500 executes the selected front end 502 locally at the client node 500. The front end

(client) programs open a TCP/IP connection back to the server to initiate message passing in order to run the applications. The front end program 502 is implemented entirely in Java which facilitates instances of client/server applications which can run concurrently on a set of multi-platform clients. The server 520 is able to send a front end program 502 to any client node 500 which has the Java runtime installed on the computer. Server policies will not involve the clients. The policies will focus on the server's control of its local resources (col. 18, ls. 9-32).

Cox et al provide two program files for each configurable application program which are provided to a network server station which operates as an on-demand server for software deployment and may also act as the application server. The on-demand server makes the first, or configuration manager, program available to an administrator classified user (either remotely at a client station or at a direct interface to the server station) to provide an interface to establish preferences for the configurable preferences of the application program which have been designated as administrator only settable. The on-demand server also provides a second, or application launcher, program to client stations on the network and served by the on-demand server. The application launcher program not only provides for a user interface to execute the application program itself but also allows a user to specify one or more of the configurable parameters of the application program. An icon is displayed on the screen of the client station through which a user may initiate execution of the application program (col. 3, ls. 50 - col. 4, l. 2).

The Examiner relies in particular on the disclosure in Cox et al that, at block 54, the configuration manager applet is distributed to one of the supported client stations 24, 24'. Alternatively, the configuration manager applet may be installed only locally at server 22 in

which case an administrator may only access the configuration manager to update preferences from the user interface of server 22 (col. 8, ls. 48-54).

The Examiner also relies on the disclosure in Cox et al that the designation of user versus administrator settable preferences may be established by the software designer providing the configurable application program. This may be accomplished, for example, by only providing a user interface to a limited number of preferences in the application launcher applet distributed as part of the application file packet (col. 9, ls. 5-11).

It is respectfully submitted that the claimed invention is not obvious over Gish in view of Cox et al. The Examiner admits the system disclosed by Gish fails to disclose specifically means wherein the client is an administrator and the at least one first software package allows the server on which the at least one first software program is stored to be administered by the client administrator. Cox et al fails to remedy this deficiency in Gish. Cox et al merely deals with certain administrator only settable configurable preferences and the ability of the user to specify one or more of the configurable parameters of the application program. Cox et al cannot be properly combined with Gish to suggest the present invention in which the distributed server administrator is scalable and version independent and can administer different versions of disparate server systems without having to correspondingly update the network client administration system.

B. Claims 5-7, 10, 12, 16 and 17 - Gish, Cox and Muschett et al.:

Claims 5-7, 10, 12, 16 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gish and Cox et al as applied to claims 1, 2, 3, 9, 11 and 14, respectively, and further in view of Muschett et al. **This rejection is respectfully traversed.**

Regarding the rejected claims, although the Examiner admits the systems disclosed by Gish and Cox et al fail to disclose means wherein the at least one first software package and the at least one second software packages are JAR files, the Examiner contends these features are well known in the art and would have been an obvious modification of the system disclosed by Gish and Cox et al, as evidenced by Muschett et al. Muschett et al is cited as disclosing a system for distributing configuration files and applications to clients wherein the at least one first software package and the at least one second software packages are JAR files (column 10, lines 46-48).

Muschett et al discloses a method and system in a computer network for dynamically bundling and launching hypertext files within archive files, wherein the computer network includes at least one client connectable to one or more servers. Initially, an archive file is established within the computer network wherein particular subroutines are maintained. Thereafter, a particular hypertext file is associated with a selected subroutine maintained within the archive file. The particular hypertext file and the selected subroutine are subsequently bundled together within the archive file. Thereafter, the archive file is automatically transmitted to the client from a server maintained within the computer network, in response to a client request to download the hypertext file, such that the subroutine and the hypertext file are downloaded within a single selected archive file from the server, thereby reducing download time and increasing data packaging efficiency. The subroutine may be composed of an applet and the hypertext file may be based on an HTML file having tags which point to particular applets and data associated with the applets (col. 4, ls. 9-28).

The Examiner particularly points to Fig. 9. As depicted at block 210, the process is initiated. As indicated at block 212, a JAR file is established within a computer network, such as,

for example, the client/server network depicted in Fig. 5. Particular applets are designated and bundled within the JAR file, as illustrated at block 214. As described at block 216, a particular driver HTML file is established, associated with the JAR file, as indicated at block 218. The HTML file, the applets and any associated resources associated with the applets, are bundled within the JAR file, as depicted at block 220. The driver HTML file contains a tag which references itself as the JAR file necessary to load the applet, in response to a client request (col. 10, ls. 39-50).

Muschett et al clearly fails to rectify the deficiencies of Gish and Cox et al. Muschett et al appears to be cited merely because it references JAR files. JAR files are per se well known and this is all Muschett et al adds to the Examiner's flawed obviousness argument.

Conclusion

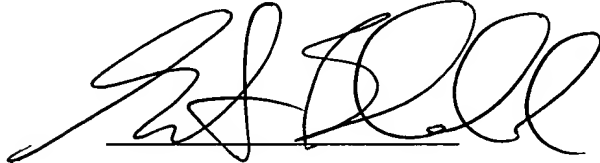
In view of the foregoing, early and favorable action is respectfully requested, as is the grant of a U.S. Patent.

It is believed that no additional claims fees are due or owing in regard to the submission of this Response and Amendment and its attached and related papers. However, if such fees are deemed due, the Office is invited to contact the undersigned at the address and telephone number listed below. A Petition for a three (3) month extension of time together with appropriate fees \$930.00 has been submitted herewith, a grant of the Petition is earnestly requested..

If any additional fees are deemed due or necessary, the Commissioner is hereby authorized to charge any fees due in connection with the present Amendment to Deposit Account 19-4293.

It is believed that a telephonic or in-person interview will in any way expedite examination proceedings related to this application, the Examiner is invited to contact the undersigned attorney of record.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Erik B. Cherdak', written over a horizontal line.

Erik B. Cherdak
Reg. No. 39,936

STEPTOE & JOHNSON LLP
1330 Connecticut Ave., NW
Washington, DC 20036
Tel: 202-429-6748

Date: 6/19/03

APPENDIX

IN THE SPECIFICATION:

Page 6, replace the last full paragraph at lines 17-25 as follows:

--To illustrate salient features of the present invention, reference first is made to a system (and corresponding methods) for facilitating distributed server administration of server systems that are scalable and version independent according to a preferred embodiment of the present invention. In particular, referring now to FIG. 1, a system 100 includes server(s) 102 configured to store and serve software packages and to be accessed by a client administrator 104 via a network 114, and an [admin] administrative server 106.--

Page 7, replace lines 1-11 as follows:

--based on a server software package or packages (e.g., a mail server system is one that runs a mail server software package to facilitate electronic mail services, etc.). As shown, multiple servers can be connected to the same network and could each be running different versions of server software (as designated V.1 and V.2). Each server must be maintained and administered (e.g., version controlled, updated, fixed, addressed for performance auditing, etc.). Therefore, a client administrator system 104 is one that is configured to operate within a client-side data processing system (e.g., a client application) and is used to access and administer each server 102 via network 114.--

Page 8, replaces lines 1-16 as follows:

reference(s) to an invocation class [114] 112, a reference to a JAR file containing this class (also known as the primary JAR file), and the network address (e.g., uniform resource locator, IP address, etc.) of the JAR file. The network address could be any address accessible via standard protocols, such as, TCP/IP, IP, IPX, etc., and could be located on a separate server, such as [admin] administrative server 106, as shown. Each reference could be stored in a common directory on server 102, such as configuration directory server 108 as shown.

Once the client administrator 104 receives the aforementioned references, it analyzes the data in the references. The client administrator 104 is configured to first look locally for the primary JAR file referenced by the server 102. If the primary JAR file is not locally installed, client administrator 104 will access the network address referenced, for example, a network address located on [admin] administrative server 106, and retrieve the primary JAR file using a network data transfer protocol, such as HTTP or FTP. Client administrator 104 is capable of installing and storing the primary JAR file locally in a directory, for example, //java/jars. When JAR files are installed locally, the files can be constructed using ISO two-character language code of the current Locale object of the Java virtual machine. For example, in the US Locale, if the primary JAR were named "dirserv40.jar," the localized JAR name would be constructed as "dirserv40_En.jar." If this localized JAR exists, it is downloaded and stored in a JAR directory.-

Page 9, replace the first full paragraph at lines 3-14 as follows:

--Referring to FIG. 2, the manifest file 200 contains include directives 202 and 204. Each include directive in manifest file 200 specifies other JAR file(s) whose contents are to be included in the runtime class loader environment of the SMUI. As described above with regard to the primary JAR file, the client administrator 104 can download each JAR file corresponding to include directives 202 and 204; if a JAR file is already locally installed, for instance, in the //java/jars/ directory, then the localized JAR file will be used instead of downloading a remotely located JAR file. Otherwise, client administrator 104 will locate the file, either on server 102, [admin] administrative server 106 or from another network address if referenced, as previously described.--

Page 11, replace lines 1-15 as follows:

--arrangement 302, a data storage subsystem 304, and multimedia input and output device(s) 306. ADP 300 may be configurable to operate as a server 102, as a client administrator 104, and as [a admin] an administrative server 106. As such, ADP 300 is configurable to store and serve software packages and other computer files (for example, JAR files, Java class references, etc.), such as those stored and served by configuration directory server 108, and to execute a server software package (for example, as designated by V.1 and V.2, as shown in FIG. 1). ADP 300 is also configurable to access a network address, download, install and process JAR files, as described in reference to client administrator 104 and FIG. 1. The arrangement and configuration of ADP 300 to operate as a server 102, as a client administrator 104, and as [a admin] an administrative server 106 will be immediately understood by those skilled in the art after reviewing the present invention and this patent document.

Page 12, replace lines 1-13 as follows:

--otherwise coupled to a network, such as network 114, and capable of accessing the server to be selected via the network. As described above, the network may be any computer or communications network, such as the Internet and World Wide Web, an intranet, Ethernet, etc. As described above in reference to FIG. 1, the server should contain references to the invocation class, the JAR file containing the invocation class (primary JAR file), and the location of the JAR file. The primary JAR file can be located at a specific network address, such as an IP address, and/or may be located on a specific server, such as server 102 or [admin] administrative server 106 described in reference to FIG. 1. The references could also be stored on the server in a common directory, such as described above with reference to FIG. 1.--

IN THE CLAIMS:

1. (currently amended) A distributed server administration system, comprising:
 - a server configured to be accessed via an electronic data network and to store and serve at least one first software package via said electronic data network, wherein said at least one first [first one] software package corresponds to at least one software system of said server; and
 - a client administrator configured to access said server, to receive said at least one first software package, and to execute said at least one first software package in conjunction with said corresponding at least one software system of said server, the execution of said at least one first

software package allowing said server to be administered by said client administrator via said electronic data network.

2. (currently amended) The system according to claim 1, further comprising

an administrative server configured to store and serve a plurality of second software packages, wherein said at least one first software package contains a reference corresponding to at least one second software package from said plurality of second software packages, at least one second software package corresponding to said software system of said server, and wherein said client administrator is further configured to access said [admin] administrative server based on said reference, to receive said at least one second software package, and to execute said at least one second software package in conjunction with said at least one software system of said server, said execution of said at least one first software package and said second software package allowing said server to be administered by said client administrator via said electronic data network.

3. (original) The system according to claim 1, wherein said server is further configured to store and serve at least one reference to a network address of a second software package, wherein said second software package corresponds to said at least one software system of said server, and said client administrator is further configured to receive said at least one network address, to access said at least one network address, to receive said second software package, and to execute said second software package in conjunction with said at least one software system of said server, said execution of said at least one first software package and said second software package

allowing said server to be administered by said client administrator via said electronic data network.

4. (original) The system according to claim 1, wherein said client administrator is further configured to install and store said at least one first software package.

5. (original) The system according to claim 1, wherein said at least one first software package is a JAR file.

6. (currently amended) The system according to claim 2, wherein said at least one first software package and said at least one second software [packages] package are JAR files.

7. (original) The system according to claim 3, wherein said at least one first software package and said second software package are JAR files.

8. (original) The system according to claim 3, wherein said client administrator is further configured to install and store said at least one first software package and said second software package.

9. (original) The system according to claim 3, wherein said second software package contains a reference to a plurality of software packages corresponding to said at least one software system of said server, and said client administrator is further configured to receive and to execute said plurality of software packages in conjunction with said at least one software system of said server.

10. (original) The system according to claim 9, wherein said plurality of software packages are JAR files and are executed together forming a Java user interface containing an administration interface to said server.

11. (original) A distributed server administration system, comprising:

a plurality of servers coupled to an electronic data network, each server of said plurality of servers being configured to store and to serve at least one first software package, said at least one first software package corresponding to the server on which said at least one first software program is stored; and

a client administrator configured to access each server in said plurality of servers via said electronic data network, to receive said at least one first software package from each server, and to execute said at least one first software package in conjunction with each server, said at least one first software package allowing said server on which said at least one first software program is stored to be administered by said client administrator.

12. (original) The system according to claim 11, wherein said at least one first software package is a JAR file.

13. (original) The system according to claim 11, wherein said at least one first software package references at least one second software package, said at least one second software package corresponds to the server on which said at least one first software program is stored and is located on a server within said plurality of servers, and said client administrator is further configured to receive said at least one second software package and to execute said at least one second software package in conjunction with said at least one first software package to allow the administration of said server on which said at least one first software program is stored.

14. (original) A method for facilitating a distributed server administration system comprising the steps of:

at a client administrator, selecting and accessing a server to be administered via an electronic data network, said server configured to store and to serve at least one first software package corresponding to said server;

at said client administrator, receiving said at least one first software package; and

at said client administrator, executing said at least one first software package in conjunction with said server to produce an administration interface wherein said server may be administered via said electronic data network.

15. (original) The method according to claim 14, wherein said at least one first software package references at least one second software package, said second software package corresponding to said server, further comprising the steps of:

at said client administrator, receiving said at least one second software package based on said reference; and

at said client administrator, executing said at least one second software package in conjunction with said at least one first software package to produce an administration interface wherein said server may be administered.

17. (original) The method according to claim 14, wherein said at least one first software package is a JAR file.

17. (currently amended) The method according to claim [14] 15, wherein said at least one first software package and said at least one second software package are JAR files.

19. (original) The method according to claim 14, wherein said client administrator is further configured to install and store said at least one first software package.

19. (currently amended) The method according to claim [18] 15, wherein said client administrator is further configured to install and store said at least one first software package and said at least one second software package.

IN THE CLAIMS:

1. (currently amended) A distributed server administration system, comprising:

a server configured to be accessed via an electronic data network and to store and serve at least one first software package via said electronic data network, wherein said at least one first [first one] software package corresponds to at least one software system of said server; and

a client administrator configured to access said server, to receive said at least one first software package, and to execute said at least one first software package in conjunction with said corresponding at least one software system of said server, the execution of said at least one first software package allowing said server to be administered by said client administrator via said electronic data network.

2. (currently amended) The system according to claim 1, further comprising

an administrative server configured to store and serve a plurality of second software packages, wherein said at least one first software package contains a reference corresponding to

at least one second software package from said plurality of second software packages, at least one second software package corresponding to said software system of said server, and wherein said client administrator is further configured to access said [admin] administrative server based on said reference, to receive said at least one second software package, and to execute said at least one second software package in conjunction with said at least one software system of said server, said execution of said at least one first software package and said second software package allowing said server to be administered by said client administrator via said electronic data network.

3. (original) The system according to claim 1, wherein said server is further configured to store and serve at least one reference to a network address of a second software package, wherein said second software package corresponds to said at least one software system of said server, and said client administrator is further configured to receive said at least one network address, to access said at least one network address, to receive said second software package, and to execute said second software package in conjunction with said at least one software system of said server, said execution of said at least one first software package and said second software package allowing said server to be administered by said client administrator via said electronic data network.

4. (original) The system according to claim 1, wherein said client administrator is further configured to install and store said at least one first software package.

5. (original) The system according to claim 1, wherein said at least one first software package is a JAR file.

6. (currently amended) The system according to claim 2, wherein said at least one first software package and said at least one second software [packages] package are JAR files.

7. (original) The system according to claim 3, wherein said at least one first software package and said second software package are JAR files.

8. (original) The system according to claim 3, wherein said client administrator is further configured to install and store said at least one first software package and said second software package.

9. (original) The system according to claim 3, wherein said second software package contains a reference to a plurality of software packages corresponding to said at least one software system of said server, and said client administrator is further configured to receive and to execute said plurality of software packages in conjunction with said at least one software system of said server.

10. (original) The system according to claim 9, wherein said plurality of software packages are JAR files and are executed together forming a Java user interface containing an administration interface to said server.

11. (original) A distributed server administration system, comprising:

a plurality of servers coupled to an electronic data network, each server of said plurality of servers being configured to store and to serve at least one first software package, said at least one first software package corresponding to the server on which said at least one first software program is stored; and

a client administrator configured to access each server in said plurality of servers via said electronic data network, to receive said at least one first software package from each server, and to execute said at least one first software package in conjunction with each server, said at least one first software package allowing said server on which said at least one first software program is stored to be administered by said client administrator.

12. (original) The system according to claim 11, wherein said at least one first software package is a JAR file.

13. (original) The system according to claim 11, wherein said at least one first software package references at least one second software package, said at least one second software package corresponds to the server on which said at least one first software program is stored and is located on a server within said plurality of servers, and said client administrator is further configured to receive said at least one second software package and to execute said at least one second software package in conjunction with said at least one first software package to allow the administration of said server on which said at least one first software program is stored.

14. (original) A method for facilitating a distributed server administration system comprising the steps of:

at a client administrator, selecting and accessing a server to be administered via an electronic data network, said server configured to store and to serve at least one first software package corresponding to said server;

at said client administrator, receiving said at least one first software package; and

at said client administrator, executing said at least one first software package in conjunction with said server to produce an administration interface wherein said server may be administered via said electronic data network.

15. (original) The method according to claim 14, wherein said at least one first software package references at least one second software package, said second software package corresponding to said server, further comprising the steps of:

at said client administrator, receiving said at least one second software package based on said reference; and

at said client administrator, executing said at least one second software package in conjunction with said at least one first software package to produce an administration interface wherein said server may be administered.

18. (original) The method according to claim 14, wherein said at least one first software package is a JAR file.

17. (currently amended) The method according to claim [14] 15, wherein said at least one first software package and said at least one second software package are JAR files.

20. (original) The method according to claim 14, wherein said client administrator is further configured to install and store said at least one first software package.

19. (currently amended) The method according to claim [18] 15, wherein said client administrator is further configured to install and store said at least one first software package and said at least one second software package.

ABSTRACT OF THE DISCLOSURE

System and method for facilitating distributed server administration of network server systems that are scalable and version independent. The system and method include and involve a server and a client administrator. The server is configured to store and to serve a software package that corresponds to a server software system of the server. The server is further configured to be accessed by client administrator via an electronic data network, such as the Internet and World Wide Web (WWW). The client administrator is configured to access the server, to receive the software package, and to process the software package to conform to the specifications of the server. Execution of the software package allows the server to be administered by the client administrator without requiring the client administrator to be pre-configured with knowledge of the server.